

Software Architecture Modelling And Analysis In Process Algebra



Guoxin Su

Centre of Quantum Computation & Intelligent Systems
Faculty of Engineering and Information Technology
University of Technology, Sydney

Doctor of Philosophy

2013

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged.

In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student

Acknowledgements

I am greatly indebted to my principal supervisor Professor Mingsheng Ying, for bringing me from logic to computer science and for creating a versatile academic environment, in which I have undertaken rigorous doctoral training whilst have been able to freely pursue research problems that interested me.

I sincerely appreciate my alternative supervisor Professor Chengqi Zhang, who founded QCIS that benefits every of its members including me and who constantly gave me important advices in person.

I am enormously grateful to my families, especially my wife, who always supports me unconditionally, and my parents, who offer me more than what I can ask for but demand little from me.

I am greatly thankful to the providers of my doctoral scholarships, i.e. the university and my faculty FEIT.

Many thanks to all other people who ever discussed with and enlightened me in person or in one of their seminars, in particular, Prof. Runyao Duan, A/Prof. Sanjiang Li, A/Prof. Yuan Feng, Dr Weiming Liu, Nengku Yu, Yangjia Li, Cheng Guo, and Shenggang Ying.

Contents

Contents	iii
Abstract	vi
Nomenclature	vii
1 Introduction	1
2 Literature Review	5
3 Connector-based Architecture: Scalability	14
3.1 Introduction	14
3.1.1 Background	15
3.1.2 Problem: Reusability and Scalability	17
3.2 Process Algebra \mathcal{PA}_1	17
3.3 ACDL Language	20
3.4 Architectural Semantics and Properties	22
3.5 Formal Analysis	24
3.5.1 Compositional Analysis	25
3.5.2 Type-based Analysis	26
3.5.3 Significance	29
3.6 Proof Details	30
3.7 Related Work and Discussions	33
4 Connector-based Architecture: Dynamism	36
4.1 Introduction	36

4.1.1	Background	37
4.1.2	Problem: Uncertainty of Component Numbers	38
4.2	Process Algebra \mathcal{PA}_2	39
4.3	Running Example	41
4.4	Architectural Modelling and Properties	47
4.4.1	The Semantic Model	47
4.4.2	Architectural Properties	50
4.4.3	Running Example Revisited	52
4.5	Formal Analysis	53
4.5.1	Conformance	53
4.5.2	Deadlock-freedom and Non-starvation	54
4.5.3	Conservation and Completeness (1)	56
4.5.4	Conservation and Completeness (2)	57
4.6	Proof Details	58
4.7	Related Work and Discussions	63
5	Peer-to-Peer Architecture	66
5.1	Introduction	66
5.1.1	Peer-to-Peer Architectural Description	66
5.1.2	Session Types and Session Composition	67
5.2	Process Algebra \mathcal{PA}_s	68
5.3	Two-level Session Types	74
5.3.1	Syntax and Semantics	74
5.3.2	Examples of Sessions	77
5.3.3	Role Projection	79
5.4	Type Discipline for \mathcal{PA}_s	81
5.4.1	Type System	81
5.4.2	Properties of Typing	84
5.5	Behavioural Analysis	85
5.6	Process Slicing	87
5.7	Examples and Proof Details	90
5.7.1	Complete Set of Roles for The Protocol Example	90
5.7.2	Supplemented Examples	91

5.7.3	Derivation of $\Gamma_{\text{prt}} \vdash P_{\text{broker}} \triangleright R_{\text{broker}}^{\text{all}}$	93
5.7.4	Proof Details of Theorems	95
5.8	Related Work and Discussions	102
6	Specification Merging	105
6.1	Introduction	105
6.1.1	Problem: Specification Merging	106
6.2	Preliminaries	108
6.2.1	Behavioural Models	108
6.2.2	Dynamic Predicate Logic	111
6.3	Specification Interpretation	112
6.4	The Merging Algorithm	113
6.4.1	Model Characterisation	117
6.4.2	A Merging Example	118
6.5	Related Work and Discussions	119
7	Conclusions	121
	Bibliography	125

Abstract

Description of the principal design of software-intense systems is fundamental in the diversified areas of software architecture. In the past two decades, a variety of Architecture Description Languages (ADLs) emerged and competed to be a suitable modelling medium for software architecture. The high-level syntax of these languages expresses the coarse-grained specifications and properties of interest for the systems, and their formal semantics enables a rigorous analysis of these properties. Notations from the dialect of Process Algebra (PA) are frequently employed by ADLs. The variety of ADLs and the popularity of PA in architectural description motivate the research of PA as formal notations of architectural description, independent of the high-level syntax of ADLs. One benefit of this study is to shed light on some verification techniques for the system properties that ADLs aim to deal with.

This thesis investigates two basic architectural models, which are the *connector-based* and *peer-to-peer* architectures, respectively. This thesis presents a PA-based ADL-like language and a PA-based semantic approach to describe and analyse the first architectural model, and employs session types for π -calculus to deal with the second model. The main contributions are as follows. First, this thesis conducts an *in-depth* study on the use of PA as formal notations for modelling these architectures, and presents various analytic techniques to facilitate the verification of the system properties of interest. Second, this thesis introduces the session type theory into the field of software architecture research and improves the state of the art of the theory by structuring the session description in the communication and integration levels. Last, to enhance the adequacy of PA as *ad hoc* architectural modelling notations, this thesis presents an algorithm to merge a modal extension of PA and the constraint-style specifications.

Nomenclature

Term/Symbol	Full Spelling/Meaning
ADL	Architecture Description Language
PA	Process Algebra
LTS	Labelled Transition System
MTS	Modal Transition System
$\mathcal{PA}_1, \mathcal{PA}_2, \mathcal{PA}_s$	PA notations
P, Q, R	processes in PA
α, β, γ	actions in PA
$\tilde{\alpha}.\tilde{\beta}$	sequences of actions
X, Y	process variables
$\longrightarrow, \Longrightarrow$	transitions in PA
A, A'	integrating session types for \mathcal{PA}_s
B, B'	communicating session types for \mathcal{PA}_s
\mathfrak{A}	architecture type
\mathcal{A}	architecture instance
\mathcal{G}	connector
\mathfrak{C}	component type
$\mathcal{E}\langle a \rangle$	component instance
$P_{c,i}^*$	canonical component instance
\mathcal{F}	configuration
\mathcal{F}_c^*	canonical configuration
$\mathcal{S}, \mathcal{S}_*$	LTS
\mathcal{M}	MTS
$S_{\mathcal{S}}, S_{\mathcal{S}_*}, S_{\mathcal{M}}$	the set of states of $\mathcal{S}, \mathcal{S}_*, \mathcal{M}$ (resp.)
$ini_{\mathcal{S}}, ini_{\mathcal{S}_*}, ini_{\mathcal{M}}$	the initial state of $\mathcal{S}, \mathcal{S}_*, \mathcal{M}$ (resp.)
$A_{\mathcal{S}}, A_{\mathcal{S}_*}, A_{\mathcal{M}}$	the set of actions of $\mathcal{S}, \mathcal{S}_*, \mathcal{M}$ (resp.)
$\longrightarrow_{\mathcal{S}}, \longrightarrow_{\mathcal{S}_*}, \longrightarrow_{\mathcal{M}}$	transition relations of $\mathcal{S}, \mathcal{S}_*, \mathcal{M}$ (resp.)